Identity and Access Management

OSIdM4HE Penn State Code Review
April 20th , 2012

AGENDA

- Introduction
- Central Person Registry
  - Functional Overview
  - IAM On Line
  - What Happened Behind The Scenes?
  - Code Review

INTRODUCTION

High Level Timeline for the CPR

- November 2009-August 2010
  - Gathered Stakeholder requirements.
    - ~ 176 people were interviewed or surveyed.
    - 730+ requirements were gathered.
    - Created & Published CPR Software Requirements Specification.
- June 2010 – October 2010
  - Architect and Design the CPR.

High Level Timeline for the CPR

- October 2010 – March 2012
  - Developed 88 SOAP-based Web Services and their associated Unit and Integration Tests.
  - Coded domestic and international matching algorithms.
  - Address validation and transformation software was installed and configured.
  - Rules were developed for affiliations.
  - UI for Identity Provisioning.

High Level Timeline for the CPR

- March 2012 – December 2012
  - Transitioning of Core Services to use the new CPR.
  - Develop remaining UIs and ID Card services
- January 2013 – December 2013
  - Business Identity Integration
    - Work with business partners to integrate identity and registration processes.
    - Implement Production ITIL based processes.
    - Develop Organization Level Agreements.

High Level Timeline for the CPR

- January 2013 – December 2015
  - Business Identity Integration
  - Begin Decommissioning of Old Services.

IAM Team Composition

Team Member	FTE
Principal Lead	1
Project Manager	.80
Technical Manager/Architect	1
Database Administrator	1
User Interface Developer	1
System Designer	.80
Java Programmers	3.5
Policy Designer	1
Communications	.20
Security Representative	.20
User Services Representative	.20

CENTRAL PERSON REGISTRY

Central Person Registry

A centralized person registry is a single data store that combines and consolidates identity information currently stored in separate and non-integrated sources throughout the University.

From The Identity and Access Management Final Report dated 2/18/2008

Central Person Registry

- Single authoritative source for identity information:
  - Biographical information (name, address, telephone number)
  - Key Identifiers
    - Person identifier
    - Userid
    - PSU ID Number

FUNCTIONAL OVERVIEW

CPR from a functional view …

CPR FUNCTIONAL OVERVIEW – HIGH LEVEL DESIGN

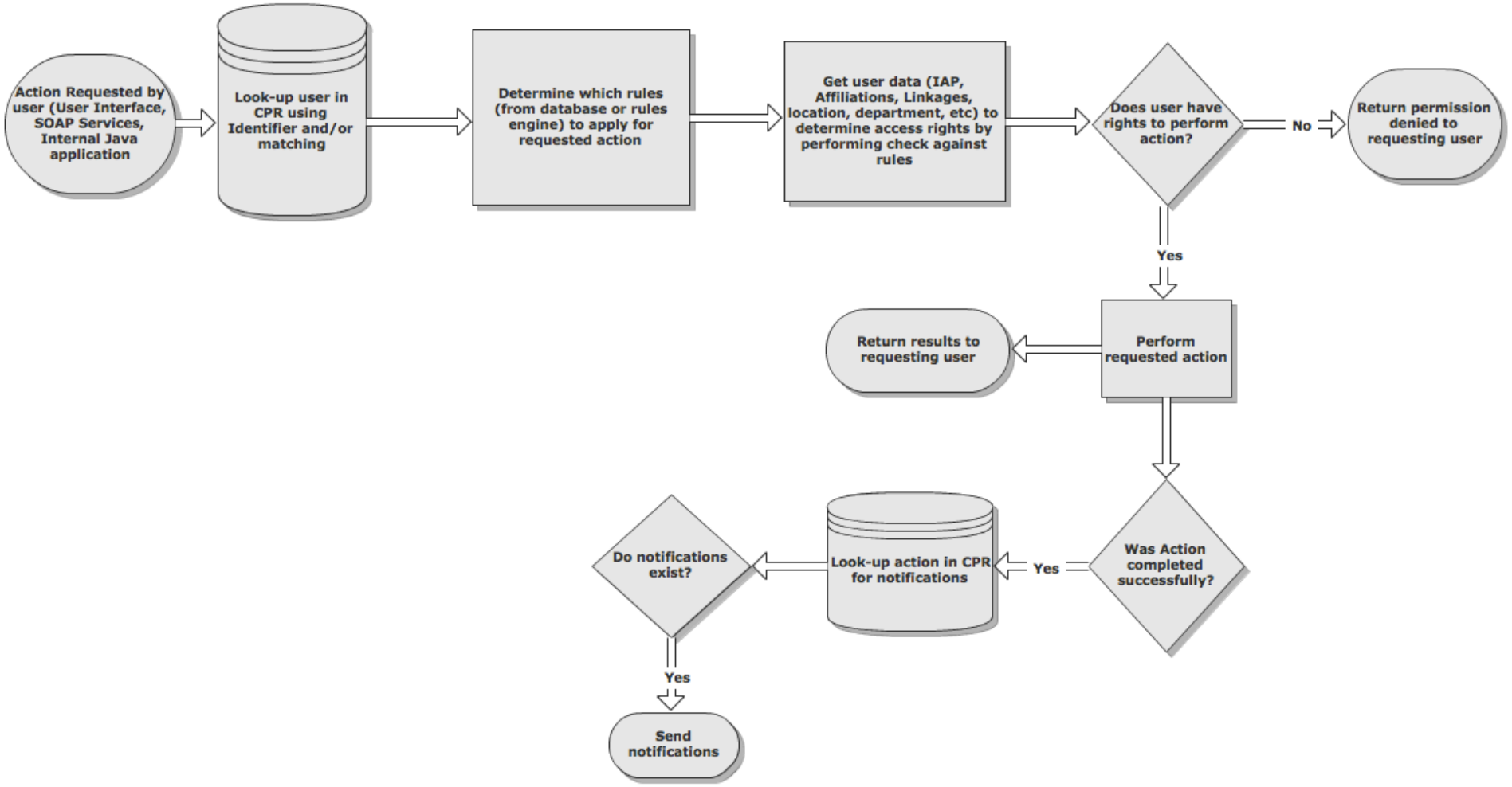
- https://wikispaces.psu.edu/display/IAM/CPR+High+Level+Design
- REGISTRY DATABASE
  - Unique Identifiers
  - Biographical Information
  - Affiliations
  - Linkages
- Identity Assurance Profiles
- Web Services
- Matching
- Service Provisioners
- Data Views
- Rules Engine

PENNSTATE

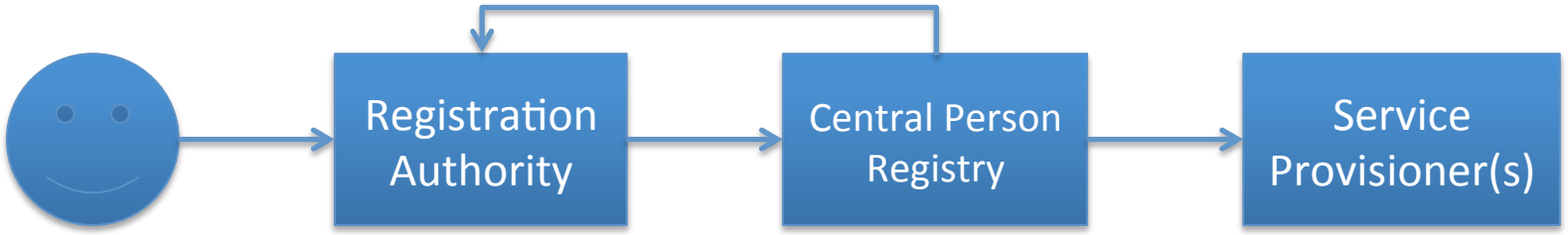
Identity and Access Management

FUNCTIONAL OVERVIEW

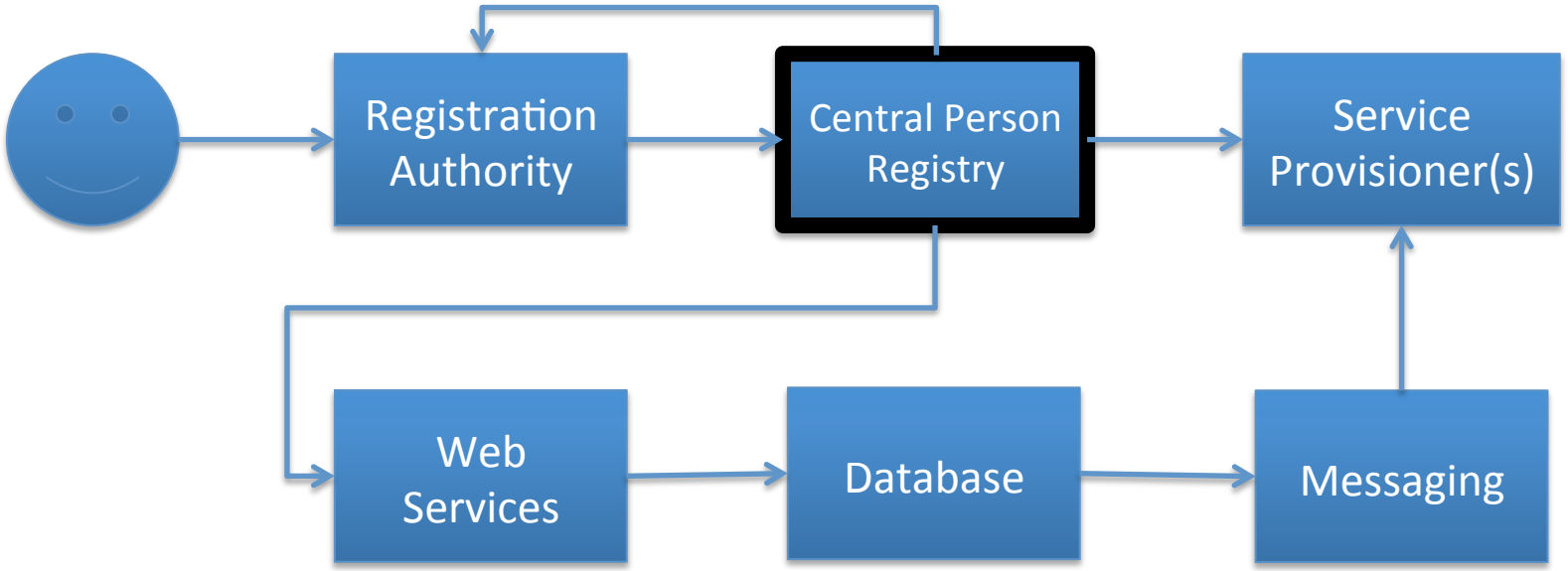
https://wikispaces.psu.edu/display/IAM/CPR+User+Request+FloW



Simple Data Flow



Simple Data Flow - Expanded



Data Model

- Addresses
- Affiliations
- Comments
- Confidentiality
- Data Types
- Date of Birth
- Email
- Gender
- IAP
- ID Card
- Names
- Person
- Person Linkage
- Phones
- Provisioning
- PSU ID
- Registration Authorities
- Userid

Data Types

- Names
  - Legal and Preferred.
- Addresses
  - Permanent, Local, Documented, Work, Billing Academic, and Billing Administrative.
- Phones
  - Permanent, Local and Work.

Data Types

- Email Address
  - University and Other.
- Document Types
  - Passport, State Driver's License, State ID Card and Military ID Card.

Service Model

- All services are SOAP-based POJOs.
- Each service has an implementation and a Service Endpoint Interface (SEI).
- Example WSDL:
  - https://apps.iam.psu.edu/Names/services/NamesPort?wsdl
- Service Documentation

Service Model

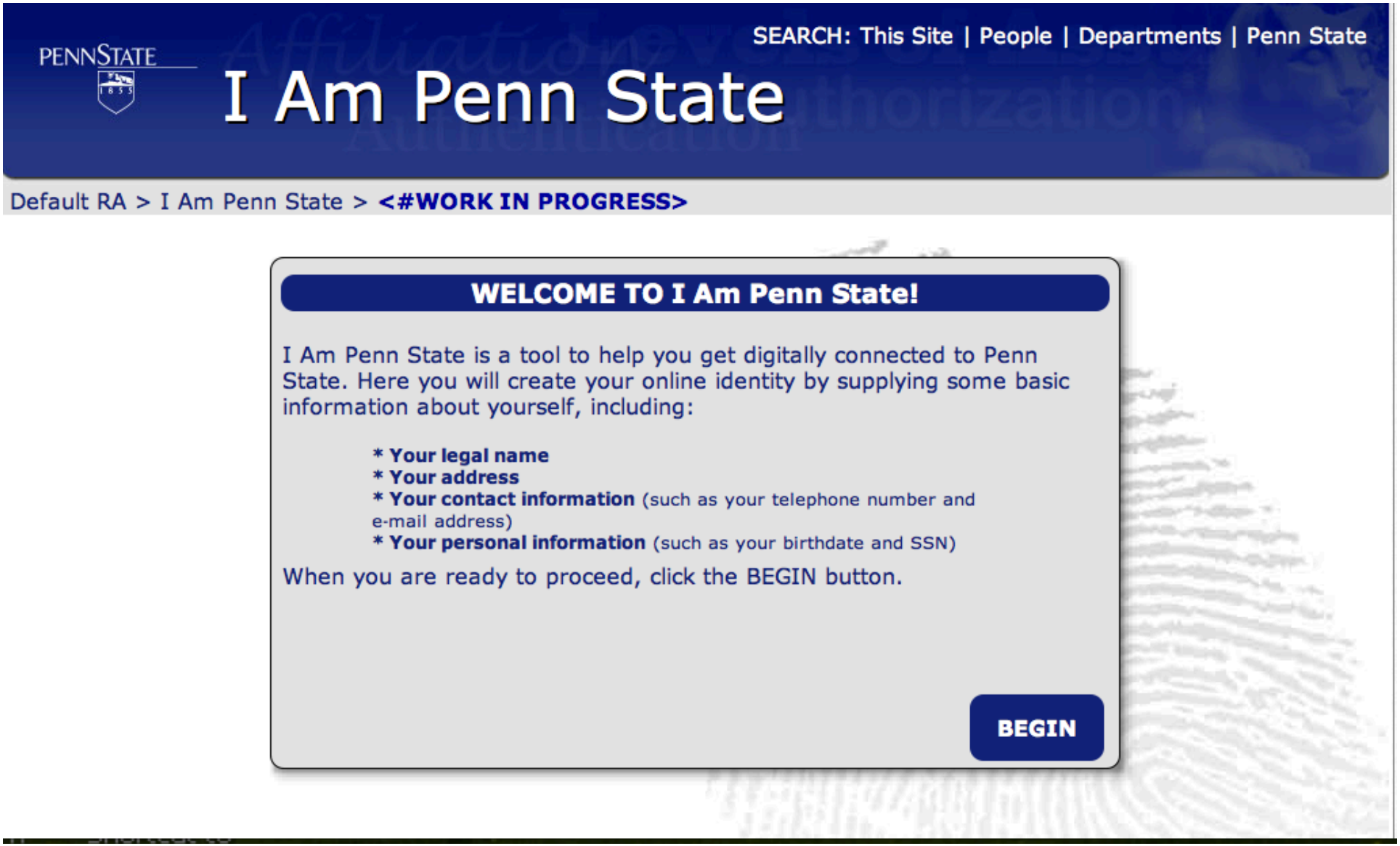
- Address
- Affiliation
- Confidentiality
- Email Address
- Find Person
- IAP
- ID Card
- Logging
- Names
- Person
- Person Linkage
- Phones
- Photo
- Rules
- Security Action
- Transform
- User Comments
- Userid

IAM ONLINE …

Demo of the IAM OnLine Identity Provisioning process ….

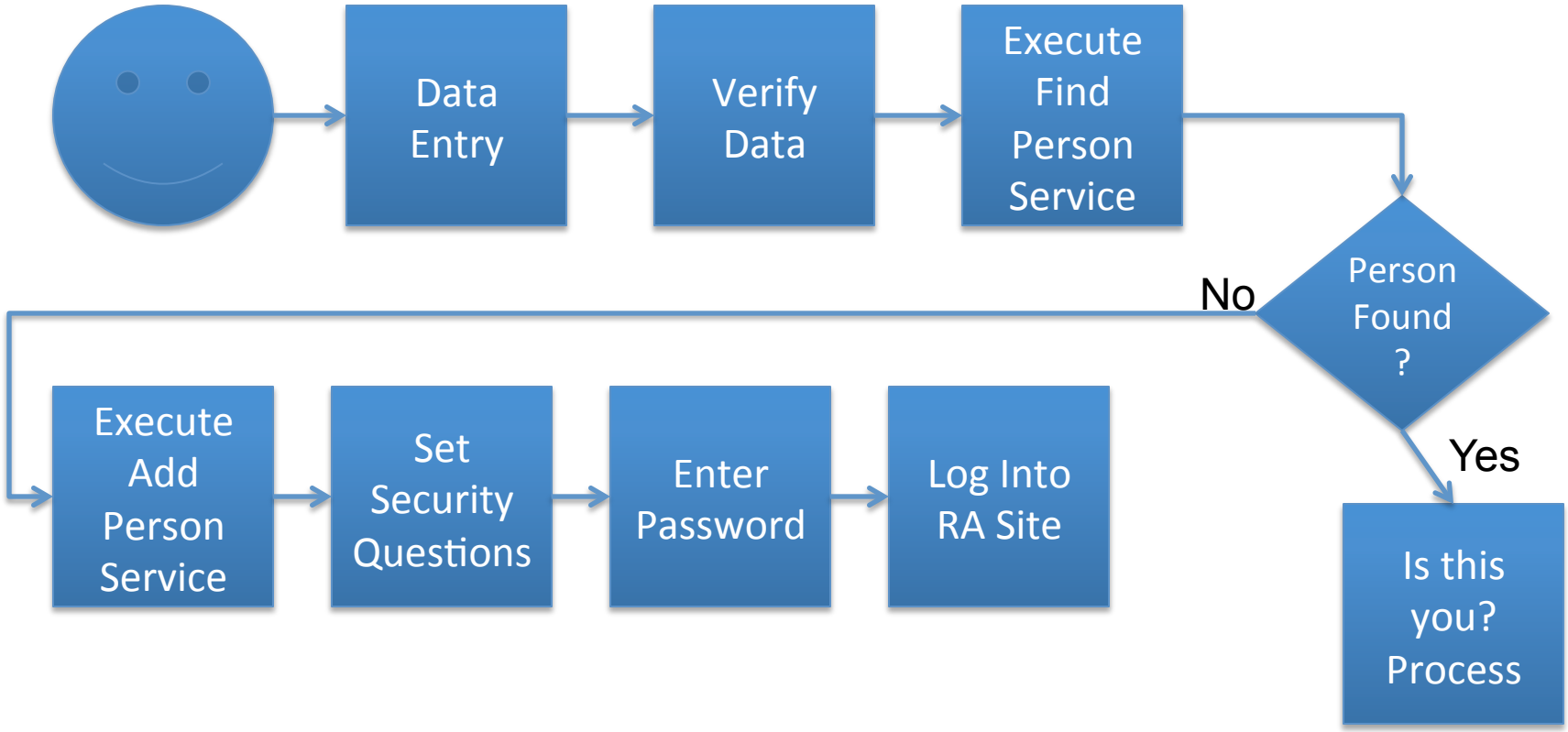
IAM ON LINE – IDENTITY PROVISIONING

- 130+ requirements documented from interviews – SII, SOS, User Services, Privacy Office, Hershey, AIT and others…
- Concept developed around Student Identity Integration project …
- Iterative process…
- Engaging more stakeholders…
- Usability


PENNSTATE
SEARCH: This Site | People | Departments | Penn State
I Am Penn State
Default RA > I Am Penn State > <#WORK IN PROGRESS>
WELCOME TO I Am Penn State!
I Am Penn State is a tool to help you get digitally connected to Penn State. Here you will create your online identity by supplying some basic information about yourself, including:
* Your legal name
* Your address
* Your contact information (such as your telephone number and e-mail address)
* Your personal information (such as your birthdate and SSN)
When you are ready to proceed, click the BEGIN button.
BEGIN

WHAT HAPPENED BEHIND THE SCENES?

Behind The Scenes



Behind The Scenes

- Service/Data Authorization using Grouper
- Find Person Service
  - Identity Match
- Add Person Service
  - Transform Address Service
  - Rules Engine
  - Messaging

Grouper

- Authorization to use a Web Service is controlled in two ways:
  - Access by the Registration Authority to call the individual services.
  - Access by the Registration Authority to perform a data manipulation (read/write) within a service.
- Initially these accesses were controlled by database tables in the CPR database, we have since moved to use Grouper.

Grouper

- The individual Registration Authorities are represented as groups within Grouper.
- The groups are also represented as role, which enables the assignment of permissions like the ability to execute a particular service.
- The Grouper Client (embedded within the CPR) is then used to determine if an RA access is valid.

Find Person Service

- Will use exact or near matching to find a person in the CPR.
- Accepts PSU ID Number, Userid, SSN, Name, Address, Date of Birth, and Gender.
- Attempts to perform an exact match using either:
  - PSU ID, Name, and Date of Birth.
  - SSN, Name, and Date of Birth.
  - Userid, Name, and Date of Birth.
- If there wasn't an exact match, it will do near matching.

Match Codes

- Help to identify duplicate or near-duplicate records.
- Match codes are encoded representations of common data types - name, street address and city.
- Match codes enable cleansing of data by reducing variations from data entry.
- e. g. name matching
  - For names William Smith, Will Smith, Bill Smith, Billy Smith.
  - Literal comparison of any two names will fail
  - Match codes generated for each of the names are the same
- Match codes generated by vendor (DataFlux) supplied product that includes region specific knowledge base(s).
- Match codes are stored in CPR database whenever name and address records are created or updated.

Match Codes

Data Element	Value	Match Code
Name	James A. Vuccolo[1]	V3W&$$$$$$$C&B_4$$$$
Address	1231 SHAMROCK AVE[2]	ZHKZ$42BY$$$$$$
City	State College	4~~3WF$$$$$$$$$

Notes:
[1] Will have the same match code for Jim A. Vuccolo, Jimmy A. Vuccolo, and so on.
[2] Will have the same match code if AVE was not specified or if Avenue was.

Identity Match (basic)

Core match logic to successfully match two identity records

- When PSU Ids of both records are available
  - PSU Ids are equal
  - Name match codes are equal
  - Birth dates are equal
- When SSNs of both records are available
  - SSNs are equal
  - Name match codes are equal
  - Birth dates are equal
- When user ids of both records are available
  - User ids are equal
  - Name match codes are equal
  - Birth dates are equal
- If no match, perform near match process

Identity Match (near match)

- Searches for potential matches performed on both current and historical data.
- Searches start with name match code.
- Data elements compared:
  - Name, street address, city and postal code, state (or country for international), birth date, and gender.
- Each comparison between records produces a match quality ranking score from 1 – 550.
- Ranking scores above a designated score (330) are classified as matches.

Identity Match (near match)

Domestic Match Criteria

Rank	Name	St	City	State	Zip	DOB
550	Y	Y	Y	Y	Y	Y
540	Y	Y	Y		Y	Y
530	Y	Y	Y	Y		Y
520	Y	Y		Y	Y	Y
510	Y	Y			Y	Y

International Match Criteria

Rank	Name	St	City	Zip	Country	DOB
550	Y	Y	Y	Y	Y	Y
540	Y	Y	Y	I	Y	Y
530	Y	Y	N	Y	Y	Y
520	Y	N	Y	Y	Y	Y
510	Y	N	Y	I	Y	Y
500	Y	Y	Y	Y	Y	P

Add Person

- Accepts data from the caller to create a new person in the registry.
- As part of its processing it performs:
  - Matching using FindPerson.
  - Address validation using TransformAddress.
  - Affiliation processing using the Rules service.
  - Based on the type of person being added, it sends messages to entities to provision services.

Transform Address

- Accepts data from the caller to validate and transform an address.
  - This function is performed using an add-on from the DataFlux product.
  - The validation is performed against either the USPS, Canadian PS or international postal services.
  - The address is then transformed into a standard format.

Rules Engine

- The CPR is using rules engine to separate the core business logic from the individual applications.
- Rules are used for the following items:
  - Affiliation transitions (internal and external).
  - Identity Assurance Profile (IAP) assignment.
  - And probably others.

Rules Engine

- The Rules Engine selected for the CPR is JBoss's Drools.
- Drools has many components, initially the CPR is using only Drools Expert, and then later it will support Drools Guvnor.
- Typically Drools is coupled with an application, we have elected to encapsulate the rules processing in a web service.

Example Rule

```
rule ”MapToEduPersonStudent”
    salience 100
    dialect "mvel”
    lock-on-active true
    when
            $known : KnownFact()
            $input : InputFact ( fact == "STUDENT_UNDERGRADUATE_CURRENT" || fact == "STUDENT_GRADUATE_CURRENT"
                    || fact == "STUDENT_MEDICAL_CURRENT" || fact == "STUDENT_LAW_CURRENT"
                    || fact == "STUDENT_UNDERGRADUATE_NONDEGREE_CURRENT" || fact ==
                            "STUDENT_GRADUATE_NONDEGREE_CURRENT"
                    || fact == "STUDENT_UNDERGRADUATE_NONDEGREE_CERTIFICATE_CURRENT"  || fact ==
                            "STUDENT_GRADUATE_NONDEGREE_CERTIFICATE_CURRENT"
                    || fact == "STUDENT_UNDERGRADUATE_PROVISIONAL_CURRENT"  || fact ==
                            "STUDENT_GRADUATE_PROVISIONAL_CURRENT" )
    then
            retract($input);
            $known.setFact("student");
            update($known);
end
```

Messaging

- Provides notification of data changes to registered entities.
- It is also used to request the provisioning and de-provisioning of services.

Messaging

- Communication between the CPR and service provisioners is accomplished using JMS (Java Messaging Service).
  - Point to point messaging is currently being used.
  - Publish and subscribe will be used in the future for notification of data changes.
- Data is encapsulated in a JMS message using JSON (JavaScript Object Notation)
  - JSON represents data using key/value pairs.

Messaging

- Apache's ActiveMQ provides the messaging infrastructure:
  - Protocols: OpenWire and STOMP.
  - All traffic is SSL encrypted.
  - AuthN/AuthZ on a per queue basis.

Sample Message (JSON Part)

```
{
        "PSU_ID":"908242211",
        "REQUESTED_BY":"jvuccolo",
        "PERSON_ID":"100000",
        "USERID":"txj229",
        "DIRECTORY_ID":"1477”,
        "NAME_TYPE":"LEGAL_NAME",
        "SERVICE_NAME":”AddName",
        "FIRST_NAME":"James",
        "MIDDLE_NAMES":null,
        "LAST_NAME":"Vuccolo",
        "SUFFIX":null
}
```

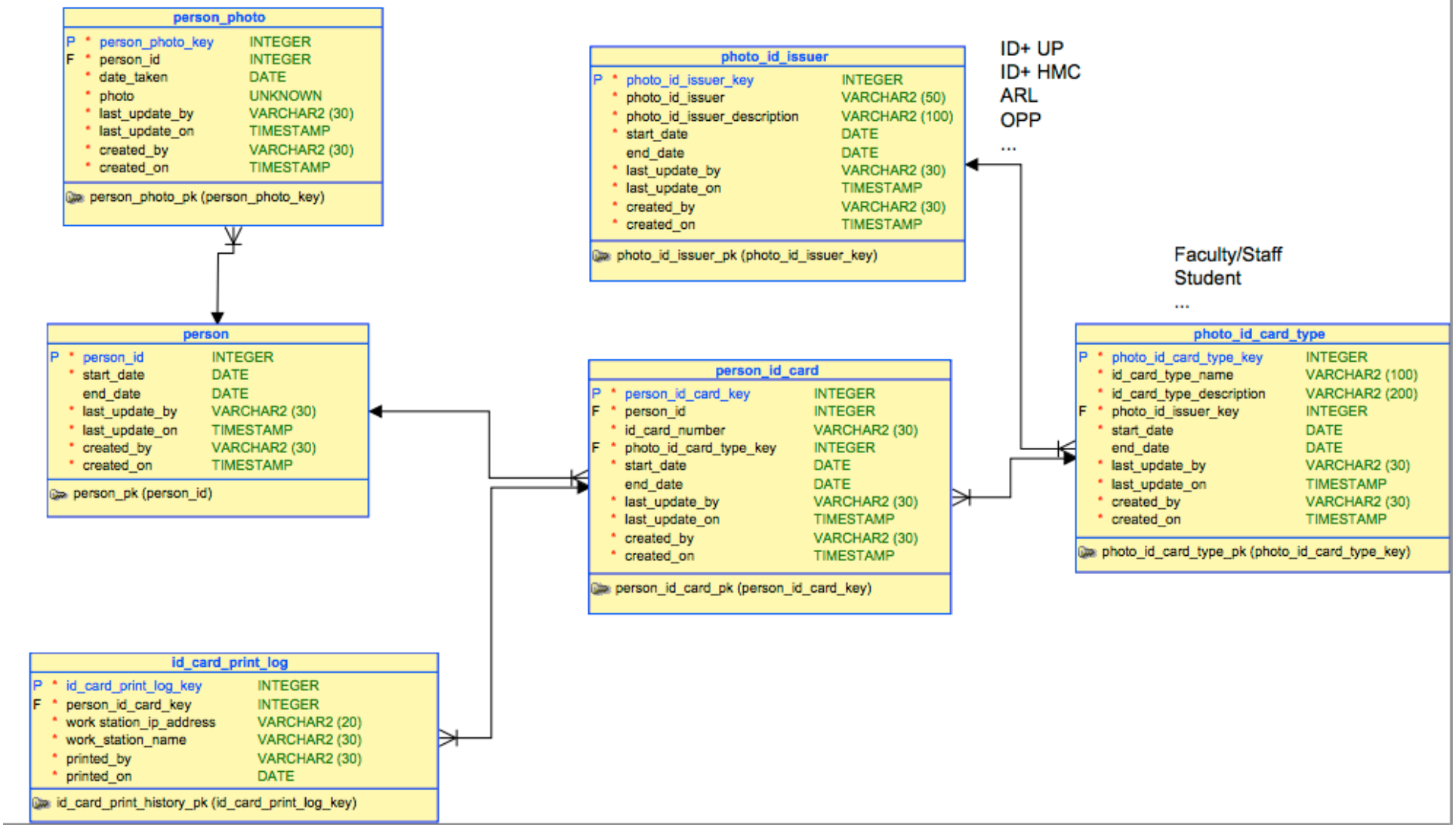
Other Registry Features

- ID Card
- Person Linking
- Identity Assurance Profiles

ID Card

- The Person Registry has the capability to store information from physical ID cards.
- Services exist to store and maintain card information.

Identity and Access Management



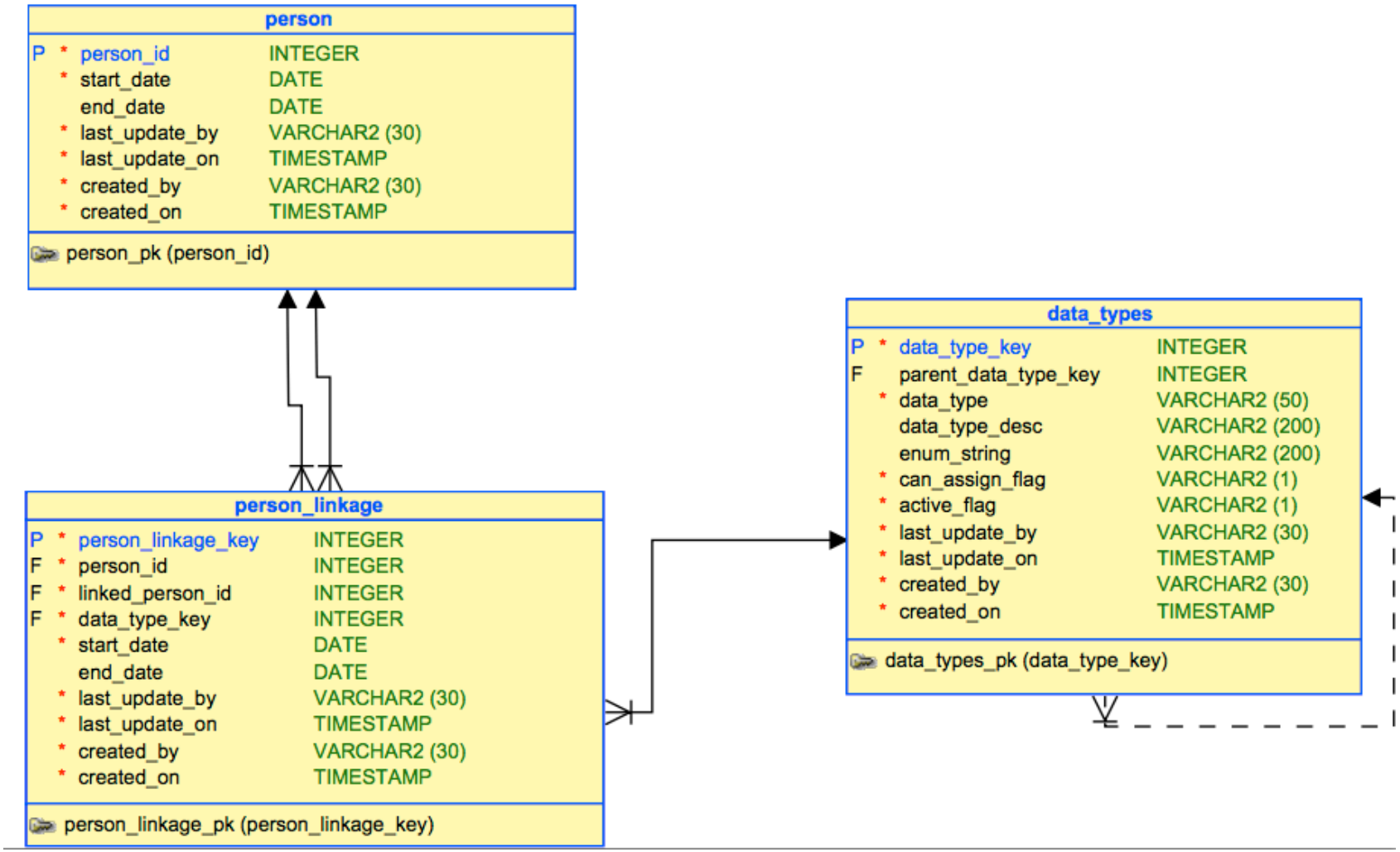
GetPhoto Service

- Will use the supplied identifier and return the JPEG photo for the person.
- Accepts PSU ID Number, Userid or Person Identifiers.
- Will return the JPEG photo.

Person Linking

- The Person Registry has the ability to establish linkages between people.
- Examples of these linkages include:
  - Dependents
  - Parent and/or Guardian
  - Spouse and/or Partner

Identity and Access Management



AddPersonLinkage

- Will use the supplied identifiers, linkage type and will establish a relationship between the two persons.
- Accepts PSU ID Number, Userid or Person Identifiers and the LinkageType.
- Will return success if the persons can be linked.

Identity Assurance Profile

- Approach
- PSU versus InCommon
- Implementation

Approach

- PSU Approach to IAP is to define our own IAPs that will map onto the InCommon IAPs.
  - This enables us to collect and verify more data than is required by InCommon.
  - This also ensures that we will still function if InCommon changes the requirements for an IAP.
    - For example, if InCommon Silver changes, Penn State's Silver will work. We would then create a new version of Penn State Silver that will map to the changed InCommon Silver.

Business Rules Summary

	PSU Tin	PSU Bronze	PSU Silver
Vetting	None	Yes (e.g.. Third Party)	In person
Proofing	None	None	In person proofing of accepted picture ID
Full Name	Self Asserted	Validate	Verify
Address of record	Self Asserted E-Mail Address	Validate physical address or E-Mail address	Verify if on ID
Date of Birth		Validate DOB (required MM/DD)	Verify
Three KB Password Q&A		Required	

Implementation

- As data is collected from various Registration Authorities, rules will be executed to determine if a user satisfies one or more IAPs.
  - Once the data and collection requirements are satisfied for a particular IAP, it will automatically be assigned.

BATCH PROCESSING

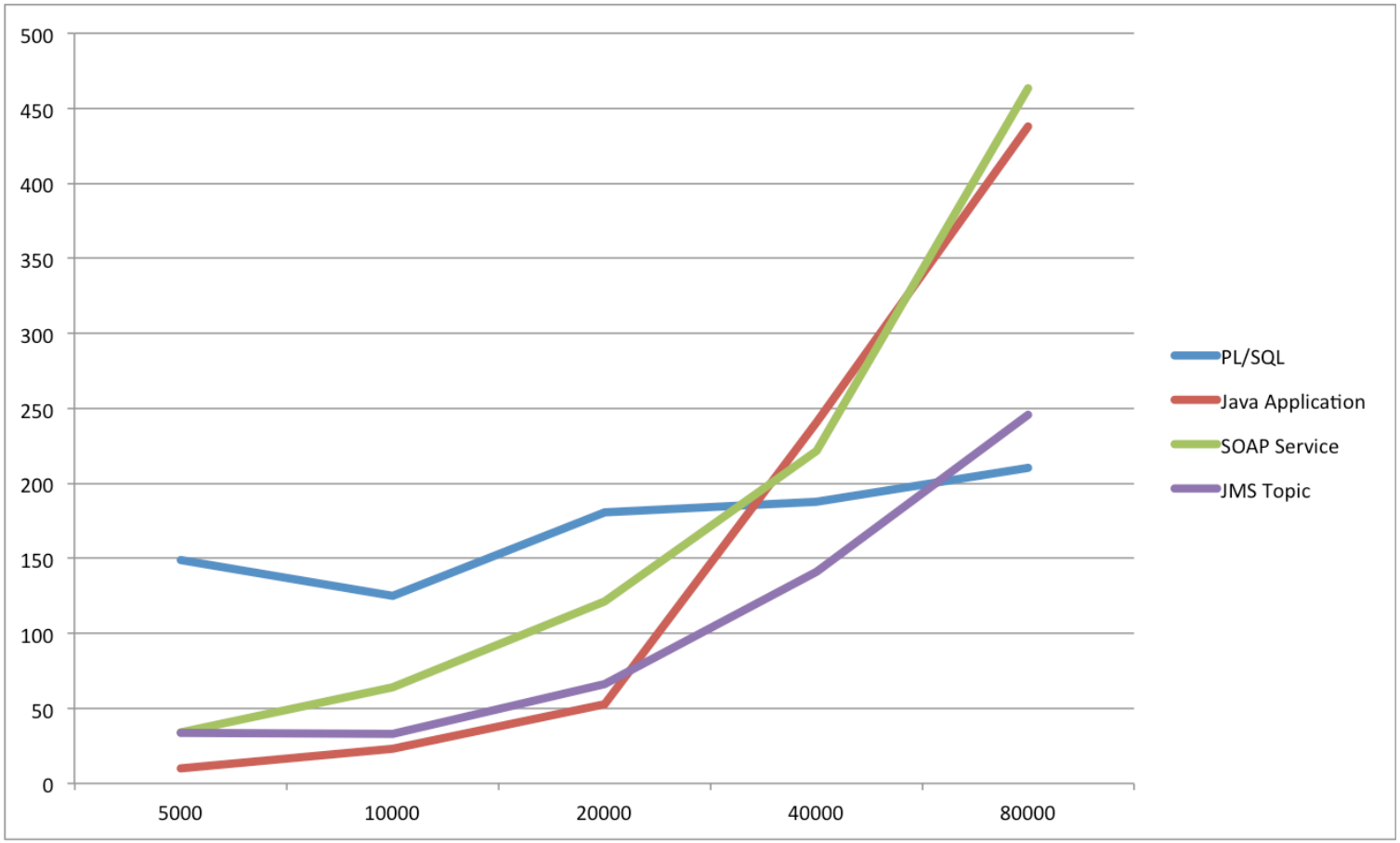
Batch Processing

- Batch processing in the past has involved the following steps:
  - Producer of raw data would place a file in a common disk space.
  - IdM system would load the data into the database and process it using one or more stored procedures.
  - The resultant file would be placed back in the common space for the producer to pick up.

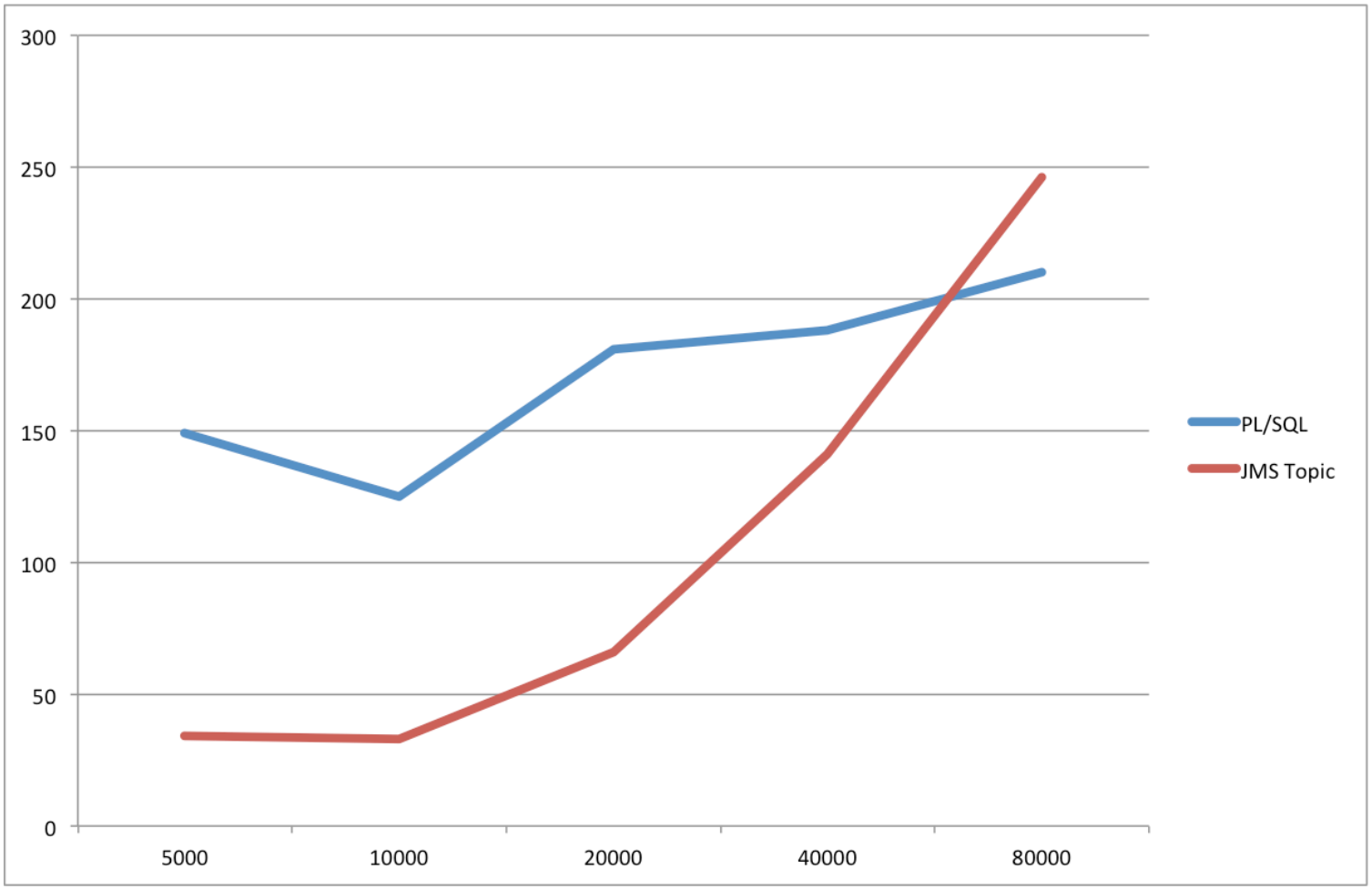
Batch Solutions

- Database Loader/Store Procedure(s)
- Java Application
- JMS Messaging Queue and/or Topic
- SOAP Service

Overall Timings



Stored Procedure vs. JMS



Our Recommended Approach

- Based on our tests, the use of a separate JMS cluster to process batch data feeds is the best solution.
- Benefits:
  - Flexibility in the type of data that can be processed by JMS
  - The use of Java enables us to use existing core logic and validation code.

CODE REVIEW

Code Review

- Javadoc
- WSDL
- Code

Javadoc

Component	URL
CPR Core Code	https://iam.psu.edu/dev_home/JavaDoc/cpr_core/
CPR Services	https://iam.psu.edu/dev_home/JavaDoc/services/

WSDL

Service	WSDL
Address	https://apps.iam.psu.edu/Address/services/AddressPort?wsdl
Affiliation	https://apps.iam.psu.edu/Affiliation/services/AffiliationPort?wsdl
Confidentiality	https://apps.iam.psu.edu/Confidentiality/services/ConfidentialityPort?wsdl
EmailAddress	https://apps.iam.psu.edu/EmailAddress/services/EmailAddressPort?wsdl
FindPerson	https://apps.iam.psu.edu/FindPerson/services/FindPersonPort?wsdl
IAP	https://apps.iam.psu.edu/IAP/services/IAPPort?wsdl
Names	https://apps.iam.psu.edu/Names/services/NamesPort?wsdl
Person	https://apps.iam.psu.edu/Person/services/PersonPort?wsdl

WSDL

Service	WSDL
PersonLinkage	https://apps.iam.psu.edu/PersonLinkage/services/PersonLinkagePort?wsdl
Phones	https://apps.iam.psu.edu/Phones/services/PhonesPort?wsdl
Rules	https://apps.iam.psu.edu/Rules/services/RulesPort?wdl
SecurityAction	https://apps.iam.psu.edu/SecurityAction/services/SecurityActionPort?wsdl
Transform	https://apps.iam.psu.edu/Transform/services/TransformPort?wsdl
UserComment	https://apps.iam.psu.edu/UserComment/services/UserCommentPort?wsdl
Userid	https://apps.iam.psu.edu/Userid/services/UseridPort?wsdl

Code

Component	URL
CPR Core Code	https://iam.psu.edu/cpr_code/cpr_core/
CPR Service Code	https://iam.psu.edu/cpr_code/services/

CPR Work Effort

Lifecycle Phase	Number of FTEs	Duration (months)
Requirements Gathering	2	10
Design	2	5
Implementation	3	15
Integration	2	?

Futures

- RESTful Web Services
- GitHub or similar repository for the CPR source code.
- JIRA
- CPR Quickstart
- API Integration Points

Contact and Community Information

- E-Mail: iam@psu.edu
- Web Site: https://iam.psu.edu/
- Follow "PennStateIAM" on:
  - Delicious
  - Twitter
  - YouTube
  - Facebook